

Amendments to the Claims

This listing of claims will replace all prior versions, and listing, of claims in the application.

1. (Currently Amended) A method for supporting a portal application responding to a request, comprising:

accepting [[the]] a request, at a container on one or more web servers, from a user that interacts with a graphical user interface (GUI) of a web application at a client side;

mapping the request to a control tree factory to generate a control tree, wherein the control tree factory is independent of the container and is accessible from multiple containers, wherein at least one of the containers is associated with a different protocol and/or a different application framework form another container;

generating [[a]] the control tree in the container by the control tree factory based on the request,[[:]] mapping the request to the control tree wherein the control tree is a logical representation of [[a]] the graphical user interface (GUI), wherein the control tree includes a set of controls, each of which represent represents one or more corresponding graphical and/or functional elements in the GUI of the web application[[s]] and [[are]] is related hierarchically to one another control in the set of controls, wherein the set of controls includes a plurality of portlets wherein each of the plurality of portlets is a self-contained application implemented on one or more web servers that renders its own GUI and is capable of communicating with another portlet of the plurality of portlets;

advancing the control tree through at least one lifecycle stage in a sequence of one or more lifecycles based on the request, wherein the set of controls in the control tree operates to interact with each other and produce a response based on the request in the at least one lifecyele stage;

providing the request response to a portlet the container that contains the at least one plurality of portlets portlet; and

aggregating the output of each of the at least one plurality of portlets and providing the output to the GUI of the web application at the client side.

2. (Canceled).

3. (Canceled).

4. (Previously Presented) The method of claim 1 wherein the step of generating a control tree comprises:

creating a metadata representation of a control tree; and
generating a class to construct the control tree based on the metadata representation.

5. (Original) The method of claim 1 wherein:

the request is a hypertext transfer protocol request (HTTP); and
the request originates from a web browser.

6. (Original) The method of claim 3, further comprising:

providing the response to a web browser.

7. (Original) The method of claim 1 wherein:

the control tree is advanced through the at least one lifecycle stage by an interchangeable lifecycle component.

8. (Original) The method of claim 1 wherein:

each one of the set of controls can have an interchangeable persistence mechanism.

9. (Original) The method of claim 1 wherein:

each one of the set of controls can render itself according to a theme.

10. (Canceled).

11. (Original) The method of claim 1 wherein:

one of the set of controls can advance through the series of at least one lifecycle stage in parallel with another of the controls.

12. (Original) The method of claim 1 wherein:

a lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.

13. (Original) The method of claim 3 wherein:

the response is an hypertext transfer protocol (HTTP) response.

14. (Original) The method of claim 1 wherein:

controls can raise events and respond to events.

15. (Original) The method of claim 1 wherein:

each one of the set of controls can be one of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.

16-29. (Canceled).

30. (Currently Amended) A machine readable storage medium having instructions stored thereon that when executed by a processor cause a system to:

accept a request, at a container on one or more web servers, from a user that interacts with a graphical user interface (GUI) of a web application at client side;

map the request to a control tree factory to generate a control tree, wherein the control tree factory is independent of the container and is accessible from multiple containers, wherein each different container is associated with different protocol and application framework;

generate [[a]] the control tree in the container based on the request,[[;]] map the request to the control tree wherein the control tree is a logical representation of [[a]] the graphical user interface (GUI), wherein the control tree includes a set of controls ,each of which represent represents one or more corresponding graphical and functional elements in the GUI of web applications and [[are]] is related hierarchically to one another, wherein the set of controls includes a plurality of portlets wherein each of the plurality of portlets is a self-contained

application implemented on one or more web servers that renders its own GUI and is capable of communicating with another portlet of the plurality of portlets;

advance the control tree through at least one lifecycle stage in a sequence of one or more lifecycles based on the request, wherein the set of controls in the control tree operates to interact with each other and produce a response based on the request ~~in the at least one lifecycle stage~~;

provide the request response to a portlet ~~the container~~ that contains the ~~at least one plurality of portlets~~ portlet; and

aggregate the output of each of the ~~at least one~~ plurality of portlets and providing the output to the GUI of the web application at client side.

31. (Canceled).

32. (Canceled).

33. (Previously Presented) The machine readable medium of claim 30, further comprising instructions that when executed cause the system to:

create a metadata representation of the control tree; and

generate a class to construct the control tree based on the metadata representation.

34. (Original) The machine readable medium of claim 30 wherein:

the request is a hypertext transfer protocol request (HTTP); and

the request originates from a web browser.

35. (Original) The machine readable medium of claim 32, further comprising instructions that when executed cause the system to:

provide the response to a web browser.

36. (Original) The machine readable medium of claim 30 wherein:

the control tree is advanced through the at least one lifecycle stage by an interchangeable lifecycle component.

37. (Original) The machine readable medium of claim 30 wherein:
each one of the set of controls can have an interchangeable persistence mechanism.
38. (Original) The machine readable medium of claim 30 wherein:
each one of the set of controls can render itself according to a theme.
39. (Canceled).
40. (Original) The machine readable medium of claim 30 wherein:
one of the set of controls can advance through the series of at least one lifecycle stage in parallel with another of the controls.
41. (Original) The machine readable medium of claim 30 wherein:
a lifecycle stage is one of: init, load state, create child controls, load, raise events, pre-render, render, save state, unload and dispose.
42. (Original) The machine readable medium of claim 32 wherein:
the response is an hypertext transfer protocol (HTTP) response.
43. (Original) The machine readable medium of claim 30 wherein:
controls can raise events and respond to events.
44. (Original) The machine readable medium of claim 30 wherein:
each one of the set of controls can be one of: Book, Page, Window, Menu, Layout, Portlet, Theme, Placeholder, Shell, LookAndFeel, Desktop, Body, Footer, Header, Head, Titlebar, ToggleButton, TreeView, TreeViewWithRadioButtons.
- 45-49. (Canceled).
50. (New) The method of claim 1, wherein:

the one or more lifecycles of the control tree is provided and managed by the container and can be modified by the container.

51. (New) The method of claim 1, wherein:

each container associates a context object with the control tree factory, wherein each context object provide access to the protocol and application framework that is associated with that container.

52. (New) The method of claim 1, wherein:

the control tree factory uses one or more meta data to construct statically created controls at initialization of the control tree, wherein dynamically created controls are created in the control tree in reaction to state, context, and events during a control tree lifecycle.